

AD-A203 309

DT

ETL-0510

A Programming Environment for Parallel Vision Algorithms Third Annual Report

Christopher Brown

**Computer Science Department
University of Rochester
Rochester, New York 14627**

July 1988

**DTIC
ELECTE
JAN 24 1989
S H D**

Approved for public release: distribution is unlimited.

Prepared for:

**U.S. Army Corps of Engineers
Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060-5546**

89 1 19 039

Destroy this report when no longer needed.
Do not return it to the originator

The findings in this report are not to be construed as an official
Department of the Army position unless so designated by other
authorized documents.

The citation in this report of trade names of commercially available
products does not constitute official endorsement or approval of the
use of such products.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S) ETL-0510		
6a. NAME OF PERFORMING ORGANIZATION University of Rochester		6b. OFFICE SYMBOL (if applicable)		7a. NAME OF MONITORING ORGANIZATION U.S. Army Engineer Topographic Laboratory	
6c. ADDRESS (City, State, and ZIP Code) Computer Science Department Rochester, New York 14627				7b. ADDRESS (City, State, and ZIP Code) CEETL-RI Fort Belvoir, VA 22060-5546	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Defense Advanced Research Projects Agency		8b. OFFICE SYMBOL (if applicable) ISTO		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DACA76-85-C-0001	
8c. ADDRESS (City, State, and ZIP Code) 1400 Wilson Boulevard Arlington, VA 22209-2308				10. SOURCE OF FUNDING NUMBERS	
				PROGRAM ELEMENT NO. 62301E	PROJECT NO.
11. TITLE (Include Security Classification) A Programming Environment for Parallel Vision Algorithms Third Annual Report					
12. PERSONAL AUTHOR(S) Christopher Brown					
13a. TYPE OF REPORT Annual		13b. TIME COVERED FROM 2/87 TO 2/88		14. DATE OF REPORT (Year, Month, Day) July 1988	
15. PAGE COUNT 17					
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Parallel Processors, Computer Vision, Butterfly Computer		
09	02				
17	08				
19. ABSTRACT (Continue on reverse if necessary and identify by block number) During the third year of the award period, the Computer Science Department of the University of Rochester concentrated on (1) operating systems, debugging support, and performance monitoring for parallel computation, (2) systems utilities for large-scale MIMD (multiple instruction stream, multiple data stream) computation, and (3) applications in active vision. This research produced internal and external reports, as well as some exportable code and several demonstration systems. Implementation of Psyche, a new operating system for large shared-memory non-uniform memory access time computers has begun. The BBN Butterfly Parallel Processor was not applied to low-level vision; instead a parallel-pipelined special-purpose device, the Datacube MaxVideo system, was integrated into the laboratory environment. The vision laboratory was also enhanced by a robot arm that positions and moves the three degree-of-freedom, two-camera robot head. Work was begun on an integrated, heterogeneously parallel system using the Butterfly, the MaxVideo, and other local computers to do complex visuo-motor tasks.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS				21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Linda H. Graff				22b. TELEPHONE (Include Area Code) (202) 355-2700	
				22c. OFFICE SYMBOL CEETL-IM-T	

Table of Contents

	Page
1. Overview	1
2. Vision	3
3. CONSUL	5
4. Psyche and Parallel Programming Environments	6
5. Performance Monitoring of Parallel Programs	7
6. Other Systems Utilities and Developments	8
7. Objectives for FY89	8
Appendix A: University of Rochester Computer Science Department DARPA-Related Publications July 1987-June 1988	10



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Annual Technical Report
DARPA Contract DACA76-85-C-0001
A Programming Environment for Parallel Vision Algorithms
Christopher Brown, P.I.
University of Rochester
13 June 1988

1. Overview

A complete set of DARPA-funded publications is appended to this summary. Key reports are placed in context below.

One goal was to create a programming environment for MIMD (Multiple instruction stream, multiple data stream) style computers. This architecture is complementary to other styles of parallel computing such as SIMD (in which identical computations are performed in parallel to different data) and neural nets.

The problem with MIMD computation, which admits multiple independent cooperating large processes and processors to run concurrently, is that the interactions between programs (for instance their data accessing) are extremely hard to monitor and even to repeat, given the potential for race conditions and the scheduling differences that can take place from run to run. Further, there are several competing, individually adequate models of parallel programs at this level. For instance, message-passing models and shared-memory models offer rather different user views of the computational resource. Although hardware can be built (like the BBN Butterfly Parallel Processor) that can efficiently support different models of parallel computation, there is a serious lack in the current state of the art of an operating system that can support several such models at once.

To improve the state of the art in programming, conceptualizing, monitoring performance, and optimizing efficiency in MIMD computation, we have been developing systems like PSYCHE (an operating system), CONSUL (a very smart autoparallelizing compiler), and MOVIOLA (a kit of performance monitoring and debugging tools.) In previous years we produced and exported about a dozen other less ambitious systems and libraries. Now, Psyche is well-specified, and implementation has begun. We expect a working kernel by January 89, and hope to integrate low levels of Psyche with the robot lab by then. CONSUL is progressing, and the interaction of the MOVIOLA debugging and performance monitoring tools have had unexpected efficacy not just in debugging but in algorithm development. The CONSUL language produces a quantified speedup, through parallelism, of between 1.9 and 18, depending on the inherent parallelism of the application.

Key Reports:

- Baldwin and Quiroz: " Parallel programming and the CONSUL..."
- Scott et al., "Design rationale for Psyche..."
- Mellor-Crummey, "Designing concurrent data structures..."
- Fowler et al., "An integrated approach"
- Marsh, "Psyche, ..."

Another goal was the acquisition of state-of-art multiprocessor. The original Butterfly Parallel Processor has certain problems that were hindering our research, which aims to transcend any particular piece of hardware. Now, a 3-node Butterfly Plus has been acquired and is on site, furnishing the primary PSYCHE development environment (enhanced by a Tektronix logic analyzer). A 24-node Butterfly Plus is also on site but not yet commissioned. We have initiated communication with MCC in Austin, TX, concerning their modular parallel experimental processor kits.

Key Reports:

LeBlanc et al., "Large-scale parallel programming..."

A third goal was to produce systems utilities for communication, file systems, and compilers. Now, several utilities are well under way. They span a broad range from parallel file systems through new languages for expressing parallel computation. Applications packages such as the current version of the neural net simulator and the image-processing utilities produced in a previous year allow speedups of up to a factor of 100 over single-workstation implementations. User interfaces to large multiprocessor computers are a difficult issue addressed by Yap's work, and we are still working to extend the range of computational models available to a user. The Ant Farm project provides capability we noticed we needed after the first DARPA Parallel Architectures Benchmark and Workshop, namely the ability to support many lightweight processes.

Key Reports:

Crowl, "A model..."

Dibble, "Bridge: ..."

Fowler and Bella, "Moviola: ..."

Goddard et al., "Rochester connectionist simulator"

LeBlanc and Jain, "Crowd Control..."

Jones, "Ant Farm:..."

Yap: "Penguin: ..."

Another goal was to commission a multiple degree-of-freedom platform for the 3-dof robot head. This work is important to test our systems concepts in a complex, visuo-motor real-time environment. Now, the PUMA 761 robot has been installed, software and hardware are almost completely integrated and debugged. Some applications (kinetic depth) and demonstrations have been programmed. The robot head has been redesigned, rebuilt, and reintegrated. PSYCHE's first application will be to manage the higher-level data structures (e.g. the world model) in an integrated parallel vision system that also uses the pipelined parallelism of the frame-rate MaxVideo image processing system.

Key Reports:

Brown, "Parallel vision ..."

Ballard and Ozcandarli, "Eye fixation..."

Our goal is to integrate hardware and software into the system using multiple types of parallelism for active vision. Now, two working groups are working (Summer 88), one on the robot/vision system, one on Psyche development, to produce an integrated demonstration by the end of the summer. The goal is to have low-level active vision

reflexes (e.g. vergence, tracking) running on our parallel pipelined low-level image processor and a Sun/3. The Butterfly Plus will be running the Psyche system. This configuration will form the basis for a continued exploration of the issues, science, and techniques behind cooperating multiple parallel computational engines.

Key Reports:

Ballard et al, "Eye movements..."
Brown, "Parallel vision ..."

Vision applications are an important part of our work, but are only indirectly supported by the contract, which views applications as potential users of the parallel systems we are developing. For example, Paul Chou's work used the Markov Random Field formulation for intermediate-level vision and produced results that have been quantified and are better than any other known techniques. We have ported his evidence-combination to the Butterfly, where it runs as a set of three cooperating agents under Tom LeBlanc's SMP system. As another example, the work of Cooper and Swain is being ported to the Connection Machine at the University of Syracuse's DARPA-funded NPAC. Object recognition, inference, quantification of performance in biologically oriented neural net computational techniques, and hardware for relaxation computations have all been under active study.

Key Reports:

Chou and Brown, "Multimodal reconstruction..."
Cooper, "Structure recognition..."
Feldman et al., "Computing with structured..."
Kyburg, "Probabilistic inference..."
Porat and Feldman, "Learning automata..."
Sher, "A Probabilistic approach..."
Simard et al., "Analysis of recurrent backpropagation"
Swain, "Object recognition..."
Swain and Cooper, "Parallel hardware..."
Watts, "Calculating the principal views..."

2. Vision

Work this year has concentrated on real-time vision hardware and algorithms. We are commissioning our MaxVideo pipelined processor, which has several independent boards each of which does a complex image-processing step, such as convolution with any of 128 arbitrary 8x8 templates, at video rates. We are now successfully tracking objects in real time, and have to used this capability in Spot, an upgrade of the Rover system described in the *1987 U. of Rochester Computer Science and Computer Engineering Research Review*. This system keeps track of multiple moving objects in the field of view.

In order to achieve the necessary flexibility and control, we have mounted the robot head on a "neck" consisting of a 6-dof robot arm. This allows us the necessary continuous motion cability, while also allowing motion in 3-D. Thus, unlike a cart, an arm makes such cabilities as viewpoint planning more relevant to robotics.

We have successfully interfaced the two eyes of our robot head to the processor, and have integrated sensory and motor function so as to do "saccades" between points of interest in the scene. Mapping of such points in three dimensions, using the robot positioner, is a current task. Substantial effort is going into understanding the individual MaxVideo boards and keeping up with software releases from the vendor. We hired a full-time staff member for vision software support, Dave Tilley (with other funds), and his help is vital in such areas as the MaxVideo board that contains an ADSP signal-processing computer.

The VME connection to the 16-processor Butterfly is now in place, and though we are experiencing some hardware difficulties the performance has been impressive. The Butterfly has now been upgraded to have floating point capability. The hardware is now physically contiguous, in our new vision lab. The plan is to write MIMD vision algorithms, using Lynx, Modula-2, or SMP, that can use the MaxVideo hardware as a powerful peripheral. This configuration will resemble in philosophy CMU's NGS architecture, but will have more power in the MIMD area (the equivalent of 16 Sun computers) and faster but less flexible power in the vision peripheral (compared to WARP).

This year Dave Sher graduated, and Paul Chou is continuing research in low-level vision. Chou's work on multi-modal evidence combination seems quite promising for real-time segmentation tasks. We have ported a parallel, asynchronous version of Paul Chou's MRF evidence combiner to the Butterfly. We can investigate the systems aspects of this implementation and also the performance of the Highest-Confidence-First algorithm (which Chou presented at IJCAI and will say more about in Miami at the IEEE vision workshop) under conditions of partial, sparse, and asynchronously arriving data.

Chou's work has produced quantifiably better algorithms for MRF optimization, which have been applied to real images for the purpose of multi-modal (intensity and depth) segmentation.

Theoretical work by Dana Ballard is leading to algorithms for kinetic depth (depth from parallax) that require basic "active visual routines" like foveal fixation, the vestibulo-ocular reflex, and other capabilities that are well-known in mammalian vision. The continuous "tracking" mode was exploited by Altan Ozcandarli, who used the MaxVideo hardware to implement Dana Ballard's kinetic depth algorithm. The system performs well, computing continuously the relative depth of points on either side of a remote fixation point that is held stable in the visual field by a low-level feedback loop between the sensors and the head effectors.

A summer project is being planned that will produce an integrated system for vision and action using several levels of hardware, and software. This exercise is meant to initiate a close collaboration in a specific project between the systems and vision/robotics communities. The highest level processing will be provided by the Butterfly Parallel Processor running the Psyche kernel being developed by the systems research group. At the lower levels there are the MaxVideo, Sun, and VAL peripherals controlling low-level vision, and body and eye movements. The particular application is just now being evolved.

3. CONSUL

The reasons why a new class of language is needed for programming multiprocessors have been set down in a paper (Baldwin, "Why We Can't Program Multiprocessors the Way We're Trying to Do It Now").

The model that indicates ways of executing CONSUL programs on multiprocessors has been refined slightly to accommodate the likely limitations of real compilers. In particular, important constraints now have "fall-back" implementations to be used in cases where a compiler is not powerful enough to generate the more efficient implementations already known. Further research will study "fall-back" implementations in more detail, quantifying their limitations, comparing them to related work from the logic programming community, etc.

A CONSUL implementation of an assignment (a simple rational arithmetic package) from one of our introductory courses has been successfully completed. This program is the first application of CONSUL to a problem not deliberately designed to demonstrate the advantages of CONSUL. A simple test of the refined execution model was also carried out by manually checking that it is able to handle the constructs appearing in this program.

Development of the CONSUL interpreter is continuing, including modifications to make it reflect the new execution model. The interpreter is able to run small programs and produce trace files from them. We have not yet tried analyzing the parallelism in these traces, although manual analysis could be done at any time.

Cesar Quiroz has done an extensive search of the existing literature on graph parsing, and has experimented with several parsing algorithms as tools for recognizing patterns in flow graphs. His results (and reactions from other interested students) are being presented in an informal seminar.

Several small programs have been executed under the interpreter. Potential parallelism, as revealed by execution traces from the interpreter, has been analyzed for these programs. Overall speed ups ranging from 1.4 to 4.9 were found. Much higher speed ups, between 4 and 18, were found in data parallel kernels within these programs. We expect higher speed-ups to be found when larger programs are run on larger inputs. A model of so-called "perfectly data parallel programs" has been devised that indicates that certain programs should exhibit linear increases in speed up with increasing input size. At least one of the sample programs conforms exactly to this model. These results are being prepared for submission to the 22nd Hawaii International Conference on System Sciences.

The interpreter has been instrumented to report statistics on its constraint satisfaction process. These statistics include the number of primitive constraints that were solved automatically (i.e., using solution heuristics coded into the interpreter), the number of primitive constraints requiring help from the user to solve, and a breakdown of primitive constraints by number of solutions to each. The first two of these numbers give some indication of the extent to which our heuristics really automate constraint satisfaction. The third is a rough indication of the size of the search spaces generated by programs. All three statistics have been encouraging for the programs run so far, although statistics reporting is too recent an addition to the system to have produced publishable numbers yet.

John Mulac is writing a CONSUL program that implements simple concurrent data base accesses. This application is a "stress test" of CONSUL as a parallel programming language, since it is an application that has in the past demonstrated substantial parallelism, but that requires fairly subtle locking protocols to exploit this parallelism correctly. Such applications are hard to write in declarative languages in which parallelism is detected automatically and programmers are not given facilities for directly describing locking protocols. Although CONSUL is such a language, Mulac's work suggests that by describing constraints on parallelism arising from the application semantics (a very natural thing to do in CONSUL) one can in fact capture most or all of the potential parallelism without needing to explicitly describe the protocols for enforcing the constraints. We plan to submit complete results from this work to this year's International Symposium on Data Bases in Parallel and Distributed Systems.

Cesar Quiroz is working out the detailed structure of his parallelism analyzer, including the data structures needed and the relationships between its various modules. He is also developing concrete examples of parallelizing rules and the flow graph patterns that enable their application. These examples and an explanation of how they are handled by the detailed analyzer will be written up as an internal working paper within a few weeks. This work should shortly lead to initial coding of parts of the analyzer.

4. Psyche and Parallel Programming Environments

"Psyche: A General-Purpose Operating System for Shared-Memory Multiprocessors," by M. L. Scott and T. J. LeBlanc, presents an overview of the Psyche philosophy and project.

Activity in the Psyche group involves directly or indirectly two faculty members and four to six graduate students actively involved. We have decided to build the operating system in two clearly- delineated layers. An interface document for the lowest, kernel, layer exists, and construction of our pilot implementation on the Butterfly is beginning. Details of the higher, supervisor layer are to be worked out concurrently with implementation of the kernel.

The kernel is expected to provide the foundation for a wide variety of future work in parallel systems. It is conceived as a lowest common denominator for a multiprocessor operating system, providing only those functions necessary to access physical resources and implement protection in higher layers. The three fundamental kernel abstractions are the segment, the address space, and the thread of control. All three are protected through capabilities. Unusual features include an inter-address-space communication mechanism based on explicit transfer of control between threads and a facility for reflecting memory protection violations upwards into user-space fault handlers. Near-term plans call for the kernel to underly both the Psyche supervisor layer and Rob Fowler's work on Metamecia.

A problem with "uniform memory access" multi-processors is that they do not scale well. Memory module contention, switch contention, and switching delay all contribute to a degradation in the effective performance of the shared memory as more processors and memory are added. On the other hand, the shared memory paradigm is a very powerful model of interprocess communication. We are therefore designing a programming model (tentatively called Metamecia) that allows the use of shared memory in a non-uniform way.

A 3-node Butterfly Plus arrived in January. This computer does not suffer from many of the memory-management limitations that beset the earlier Butterflies. A 24-node Butterfly Plus has been placed on order and is expected in April or May. Implementation of Psyche is underway on the new architecture. A target date has been set in August for the first demonstration of a working application.

5. Performance Monitoring of Parallel Programs

We are continuing our investigations of the effective implementation of parallel algorithms. The main thrust continues to be the construction of parallel performance monitoring tools and experimentation with the use of these tools.

We have constructed a set of tools for instrumenting parallel programs on the Butterfly for performance analysis. Each process in an instrumented program records on its own "history tape" each of its interactions with shared objects including the relative timing of the operations. The collection of history tapes from the individual processes can be combined to give a consistent view of the execution of the program as a whole. This view contains information useful for identifying critical paths, bottlenecks, and hot spots in the program. We are presently working on the user interface of the package to make it usable by other than its implementors.

An execution of a parallel program instrumented for performance monitoring generates a massive amount of data. This data is incomprehensible in its raw form so we are developing an interactive graphical display and analysis program called Movieola. The display and user interface aspects are currently fully functional and some simple analytic tools (critical path analysis) will be added by the end of Summer. We will expect that Movieola will also serve as the basis for the user interface of Mellor-Crummey's interactive parallel debugging tools.

The work on the "streams" package part of the NFS (Network File System) interface to the Butterfly was completed by Jonathan Payne together with Mellor-Crummey and Smithline. Mellor-Crummey began work on the integrated instrumentation package that extends Instant Replay with the performance monitoring package. This uses the streams package for asynchronous transfer of "history data", but due to problems with the Ethernet connection to the Butterfly we are not yet achieving the hoped for transfer rates. Bella continues to work on improving the user interface to Movieola. In January Smithline completed his TA and became available to the project. Fowler and Smithline began the design an integrated and extensible toolkit of debugging and performance analysis tools. The toolkit runs under the Aegis of a Lisp system and will incorporate Movieola and the instrumentation packages.

Using Movieola and the instrumentation package, we have been experimenting with their use in the debugging and performance analysis and tuning. Mellor-Crummey has been applying them to the development of parallel sorting programs. Bella has been incorporating the results of these experiences in the further development of Movieola. Fowler and Smithline completed the design of the integrated, extensible toolkit and implementation has begun. Movieola is now callable from the Lisp system and basic functionality is rapidly being added to the system. We will soon begin to extend the toolkit with analysis tools written in Lisp.

6. Other Systems Utilities and Developments

"An Empirical Study of Message-Passing Overhead," by M. L. Scott and A. L. Cox, appeared at the 7th International Conference on Distributed Computing Systems in Berlin, West Germany in September 1987. It reports on efforts to optimize the performance of the LYNX run-time support package, and presents a detailed breakdown of costs in the final implementation. This breakdown (1) reveals the marginal cost of various features of LYNX, (2) carries important implications for the costs of related features in other languages, and (3) sets an example for similar studies in other environments.

The "Ant Farm" library package is essentially complete and is now being used to develop applications. It supports extremely large numbers (c. 25,000) of lightweight processes in Modula-2 with location-transparent communication.

We have completed the construction and performance studies of the Elmwood operating system for the Butterfly. A paper describing this work entitled "Elmwood - An Object-Oriented Multiprocessor Operating System" will appear in Rochester's Computer Science and Engineering Research Review and is being submitted to Software - Practice and Experience. We expect this implementation to serve as a basis for the Psyche kernel.

Work is underway to port Lynx to run under Berkeley UNIX 4.3. The eventual goal is to unify UNIX and Butterfly implementations so that processes on the multiprocessor and on workstations can communicate transparently.

We have defined a limited NFS (Network File System) interface for the Butterfly so that *program monitoring information* can be transferred from the Butterfly to SUN workstations and disks for analysis. We are currently finishing the implementation of our NFS interface on the Butterfly.

"Crowd Control: Coordinating Processes in Parallel" by T.J. LeBlanc and S. Jain will appear in the Proc. International Conference on Parallel Processing in August. This paper describes a library package for the Butterfly that can be used to create a parallel schedule for large numbers of processes. A partial order is imposed on the execution based on an arbitrary embedding of processes in a balanced binary tree.

7. Objectives for FY89

- (1) Utilize more fully the power of the parallel pipelined image -processor. This device presents a difficult interface to the user. We hope to develop abstractions of its behavior, a stable configuration for its hardware and cable connections, and a usable, flexible library of utilities.
- (2) Continue laboratory integration. Higher-bandwidth communications between some system components are needed. Libraries are needed to let different computers initiate, monitor, and control activities in the laboratory (e.g. controlling low-level vision or robot motion from the Butterfly Plus). Scientifically, the style of distributed parallel control and its implementation for active vision must be investigated. We shall make extensive visits to other robot labs (General Electric Research, University of Oxford) to expand our understanding of the state of the art.

- (3) Develop Psyche, use it to illustrate multiple models of parallel computation on one multi-computer, develop scheduling and resource allocation packages adequate to perform active vision. Psyche will continue to develop beyond its state in Jan 89. It offers users a level of abstraction above kernel level but below applications programs, at which different styles of parallel computation (shared memory, message-passing) can be imposed upon the hardware. The active vision domain will require system services similar to some "real time" operating systems, and they can be provided at this "package" level.
- (4) Investigate techniques and science of integrating planning and sensing. This rather old topic is still central, and a working group including James Allen and his students is being started to pursue the cooperation of planning, perceiving, and acting. A domain of moving objects, perhaps toy trains, will be used.
- (5) Investigate advanced inference techniques.
- (6) Investigate physically flexible robotic hardware. Control of slightly flexible robot hardware has been investigated to some extent, but robots with very flexible members, moving loads heavier than their component links, are much less well understood.
- (7) Investigate object recognition using techniques of principal views, decision trees, relaxation computations.
- (8) Continue work in connectionist learning and vision, especially motion vision.

Appendix A

University of Rochester Computer Science Department

DARPA-Related Publications

July 1987 - June 1988

University of Rochester Computer Science Department
DARPA-Related Publications
July 1987 - June 1988

Aloimonos, J. and C.M. Brown, "Robust computation of intrinsic images from multiple cues," in Brown, C.M. (Ed). *Advances in Computer Vision* (Vol. 1). Hillsdale, NJ: Lawrence Erlbaum Assoc., Pub., 1988.

Aloimonos, J. and M.J. Swain, "The paraperspective projection," submitted, *Computer Vision, Graphics, and Image Processing*, November 1987.

Aloimonos, J. and M.J. Swain, "Shape from patterns: Regularization," CAR-TR-283, Center for Automation Research, U. Maryland, April 1987; to appear, *Int'l. J. of Computer Vision*, 1988.

Baldwin, D., "Applying the RASP-SL description language to a benchmark suite," extended abstract, *Workshop on High-Level Synthesis*, January 1988.

Baldwin, D., "Constraint description and extraction in RASP," submitted, *Int'l. Conf. on Computer-Aided Design* (November 1988), April 1988.

Baldwin, D., "Preliminary estimates of parallelism in CONSUL programs," submitted, *Hawaii Int'l. Conf. on System Sciences* (January 1989), June 1988.

Baldwin, D., "SEEDS: A wide domain digital circuit simulator," submitted, *Int'l. Conf. on Computer-Aided Design* (November 1988), April 1988.

Baldwin, D. and J. Mulac, "CONSUL as a language for describing concurrent database operations," submitted, *Int'l. Symp. on Databases in Parallel and Distributed Systems* (December 1988), May 1988.

Baldwin, D. and C.A. Quiroz, "Parallel programming and the CONSUL language," *Proc., Int'l. Conf. on Parallel Processing*, St. Charles, IL, August 1987.

Ballard, D.H., "Eye movements and spatial cognition," TR 218, Computer Science Dept., U. Rochester, November 1987; presented, *AAAI Spring Symposium Series on Physical and Biological Approaches to Computational Vision*, March 1988.

Ballard, D.H., "Eye movements and visual cognition," *Workshop on Spatial Reasoning and Multisensor Fusion*, October 1987.

Ballard, D.H., "Form perception as transformation," TR 148, Computer Science Dept., U. Rochester, October 1986; also appears as "Transformational form perception," in C.M. Brown (Ed). *Advances in Computer Vision*. Hillsdale, NJ: Lawrence Erlbaum Associates, Pub., 1988.

Ballard, D.H., "Interpolation coding: A representation for numbers in neural models," TR 175 (revised), Computer Science Dept., U. Rochester, September 1986; *Biological Cybernetics* 57, pp. 389-402, 1987.

Ballard, D.H., C.M. Brown, D.J. Coombs, and B.D. Marsh, "Eye movements and computer vision," *1987-88 Computer Science and Engineering Research Review*, Computer Science Dept., U. Rochester, October 1987.

Ballard, D.H. and A. Ozcandarli, "Eye fixation and early vision: kinetic depth," submitted, *2nd Int'l. Conf. on Computer Vision* (December 1988), May 1988.

Basu, A. and C.M. Brown, "Algorithms and hardware for efficient image smoothing," *Computer Vision, Graphics, and Image Processing* 40, 2, pp. 131-146, November 1987.

Brown, C.M. (Ed). *Advances in Computer Vision* (Vols. I and II). Hillsdale, NJ: Lawrence Erlbaum Assoc., Pub., 1988.

Brown, C.M., "Parallel vision with the ButterflyTM computer," invited paper, *Proc., Third Int'l. Conf. on Supercomputing*, May 1988.

Brown, C.M., "Progress in image understanding at the University of Rochester," *Proc., DARPA Image Understanding Workshop*, pp. 73-77, April 1988.

Chou, P.B. and C.M. Brown, "Multimodal reconstruction and segmentation with Markov Random Fields and HCF optimization," *Proc., DARPA Image Understanding Workshop*, pp. 214-221, April 1988.

Chou, P.B., C.M. Brown, and R. Raman, "A confidence-based approach to the labeling problem," *Proc., IEEE Workshop on Computer Vision*, November 1987.

Cooper, P.R., "Structure recognition by connectionist relaxation: formal analysis," *Proc., DARPA Image Understanding Workshop*, pp. 981-993, April 1988; to appear, *Proc., 1988 Canadian Artificial Intelligence Conference*, June 1988.

Crowl, L.A., "A model for parallel programming," in TR 209 (*Proc., 1988 Open House*), Computer Science Dept., U. Rochester, May 1988.

Crowl, L.A., "Shared memory multiprocessors and sequential programming languages: A case study," *Proc., 21st Hawaii Int'l. Conf. on System Sciences*, 103-108, January 1988.

Dibble, P.C., "Attacking the I/O bottleneck with parallelism," in TR 209 (*Proc., 1988 Open House*), Computer Science Dept., U. Rochester, May 1988.

Dibble, P.C., M.L. Scott, and C.S. Ellis, "Bridge: A high-performance file system for parallel processors," *Proc, 8th Int'l. Conf. on Distributed Computing Systems*, to appear, June 1988.

Feldman, J.A. and D.H. Ballard, "Connectionist models and their properties," in D. Waltz and J.A. Feldman (Eds). *Connectionist Models and their Applications*. Norwood, NJ: Ablex Publishing Company, pp. 13-62, 1988.

Feldman, J.A., M.A. Fanty, and N. Goddard, "Computing with structured neural networks," *IEEE Computer* 21, 3, pp. 91-103, 1988.

Finkel, R.A., M.L. Scott, Y. Artsy, and H.-Y. Chang, "Experience with Charlotte: Simplicity and function in a distributed operating system," *IEEE Trans. on Software Engineering* (Special Issue on Design Principles for Experimental Distributed Systems), to appear, 1988.

Fowler, R.J. and I. Bella, "Moviola: An interactive execution graph browser," in preparation, April 1988.

Fowler, R.J., T.J. LeBlanc, and J.M. Mellor-Crummey, "An integrated approach to parallel program debugging and performance analysis on large-scale multiprocessors," to appear, *Proc., ACM Workshop on Parallel and Distributed Debugging*, May 1988.

Gafter, N.M., "Algorithms and data structures for parallel incremental parsing," *Proc., Int'l. Conf. on Parallel Processing*, August 1987.

Goddard, N.H., K.J. Lynne, and T. Mintz, "Rochester connectionist simulator," TR 233, Computer Science Dept., U. Rochester, Spring 1987; revised March 1988.

Kyburg, H.E., "Full belief," TR 245, Computer Science Dept., U. Rochester, February 1988; to appear, *Theory and Decision*, 1988.

Kyburg, H.E., "Probabilistic inference," TR 230, Computer Science Dept., U. Rochester, November 1987.

Kyburg, H.E., "Probabilistic inference and probabilistic reasoning," TR 248, Computer Science Dept., U. Rochester, April 1988; submitted, *Nat'l. Conf. on Artificial Intelligence (AAAI)* (August 1988), April 1988.

LeBlanc, T.J., "Structured message passing on a shared-memory multiprocessor," *Proc., 21st Annual Hawaii Int'l. Conf. on System Sciences*, pp. 188-194, January 1988.

LeBlanc, T.J. and S. Jain, "Crowd control: Coordinating processes in parallel," *Proc., Int'l. Conf. on Parallel Processing*, pp. 81-84, August 1987.

LeBlanc, T.J., M.L. Scott, and C.M. Brown, "Large-scale parallel programming: Experience with the BBN Butterfly parallel processor," to appear, *ACM SIGPLAN Symp. on Parallel Programming: Experience with Applications, Languages, and Systems*, July 1988.

Mailloux, G.E., F. Langlois, P.Y. Simard, and M. Bertrand, "Analysis of heart motion from two-dimensional echocardiograms by velocity field decomposition," *Computer in Cardiology*, Leuven, IEEE Computer Society Press, October 1987.

Mailloux, G.E., F. Langlois, P.Y. Simard, and M. Bertrand, "Restoration of the velocity field of the heart from two-dimensional echocardiograms," submitted, *IEEE Biomedical Imaging*, October 1987.

Marsh, B.D., "Psyche: a NUMA operating system kernel," in TR 88-03 (*Proc., 3rd Annual U. Buffalo Graduate Conf. on Computer Science*), Computer Science Dept., SUNY Buffalo, March 1988.

Mellor-Crummey, J.M., "Concurrent queues: Practical fetch-and- Φ algorithms," TR 229, Computer Science Dept., U. Rochester, October 1987.

Mellor-Crummey, J.M., "Designing concurrent data structures for correctness," submitted, *IEEE Trans. Computers* (Special Issue on Parallel and Distributed Algorithms), March 1988.

Mellor-Crummey, J.M., "Experiences with the BBN Butterfly," invited paper, *Proc., COMPCON Spring '88*, 101-104, February 1988.

Mellor-Crummey, J.M., T.J. LeBlanc, L.A. Crowl, N.M. Gafter, and P.C. Dibble, "Elmwood---An object-oriented multiprocessor operating system," TR 226 and BPR 20, Computer Science Dept., U. Rochester, September 1987; *1987-88 Computer Science and Engineering Research Review*, Computer Science Dept., U. Rochester, October 1987; submitted, *Software--Practice and Experience*, September 1987.

Porat, S., "Infinite behavior in connectionist models with asymmetric weights," *1987-88 Computer Science and Engineering Research Review*, Computer Science Dept., U. Rochester, October 1987; revised version in R. Trappl (Ed). *Cybernetics and Systems '88* (Proc., 9th European Meeting on Cybernetics and Systems Research). Dordrecht: Kluwer Academic Publishers, pp. 1023-1030, 1988.

Porat, S. and J.A. Feldman, "Learning automata from ordered examples," TR 241, Computer Science Dept., U. Rochester, April 1988; extended abstract submitted, *Workshop on Computational Learning Theory* (August 1988), March 1988.

Scott, M.L. and R.A. Finkel, "A simple mechanism for type security across compilation units," *IEEE Trans. on Software Engineering*, to appear, July 1988.

Scott, M.L., T.J. LeBlanc, and B.D. Marsh, "Design rationale for Psyche, a general-purpose multiprocessor operating system," to appear, *1988 Int'l. Conf. on Parallel Processing*, August 1988.

Scott, M.L. and S.-K. Yap, "A grammar-based approach to the automatic generation of user-interface dialogues," to appear, *Proc., 1988 ACM Conf. on Human Factors in Computing Systems (CHI '88)*, May 1988.

Sher, D., "Generating robust operators from specialized ones," *Proc., IEEE Workshop on Computer Vision*, November 1987.

Sher, D., "A probabilistic approach to low-level vision," Ph.D. Thesis and TR 232, Computer Science Dept., U. Rochester, October 1987.

Sher, D., "Tunable facet model likelihood generators for boundary pixel detection," *Proc., IEEE Workshop on Computer Vision*, November 1987.

Simard, P.Y. and G.E. Mailloux, "A projection operator for the restoration of divergence-free vector fields," *IEEE Trans. PAMI* 10, 2, pp. 248-256, March 1988.

Simard, P.Y. and G.E. Mailloux, "Vector field restoration by the method of convex projections," submitted, *IEEE Trans. PAMI*, January 1988.

Simard, P.Y., M. Ottaway, and D.H. Ballard, "Analysis of recurrent backpropagation," presented, *Snowbird Conf. on Neural Networks*, April 1988.

Swain, M.J., "Object recognition from a large database," in TR 209 (*Proc., 1988 Open House*), Computer Science Dept., U. Rochester, May 1988.

Swain, M.J., "Object recognition from a large database using a decision tree," *Proc., DARPA Image Understanding Workshop*, April 1988.

Swain, M.J. and P.R. Cooper, "Parallel constraint satisfaction," TR 255, Computer Science Dept., U. Rochester, to appear, June 1988.

Swain, M.J. and P.R. Cooper, "Parallel hardware for constraint satisfaction," *Proc., DARPA Image Understanding Workshop*, pp. 620-624, April 1988; to appear, *Proc., Nat'l. Conf. on Artificial Intelligence (AAAI)*, August 1988.

Watts, N., "Calculating the principal views of a polyhedron," TR 234, Computer Science Dept., U. Rochester, December 1987; to appear, *Int'l. Conf. on Pattern Recognition*, November 1988.

Yap, S.-K., "PENGUIN: A language for specifying user interface dialogues," in TR 209 (*Proc., 1988 Open House*), Computer Science Dept., U. Rochester, May 1988.

Yap, S.-K. and A. Albicki, "Covering a set of test patterns by a cellular automaton," *1987-88 Computer Science and Engineering Research Review*, Computer Science Dept., U. Rochester, October 1987.